

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

JULIO ARTUR DEBASTIANI

SERVIDOR REST PARA EMISSÃO DE NOTA FISCAL ELETRÔNICA

MONOGRAFIA DE ESPECIALIZAÇÃO

PATO BRANCO
2017

JULIO ARTUR DEBASTIANI

SERVIDOR REST PARA EMISSÃO DE NOTA FISCAL ELETRÔNICA

Monografia de especialização apresentada na disciplina de Metodologia da Pesquisa, do Curso de Especialização em Tecnologia Java, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Profa. Beatriz Terezinha Borsoi

PATO BRANCO
2017



MINISTÉRIO DA EDUCAÇÃO
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Especialização em Tecnologia Java



TERMO DE APROVAÇÃO

SERVIDOR REST PARA EMISSÃO DE NOTA FISCAL ELETRÔNICA

por

JULIO ARTUR DEBASTIANI

Este trabalho de conclusão de curso foi apresentado em 10 de novembro de 2017, como requisito parcial para a obtenção do título de Especialista em Tecnologia Java. Após a apresentação o candidato foi arguido pela banca examinadora composta pelos professores Andreia Scariot Beulke e Vinicius Pegorini. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Beatriz Terezinha Borsoi
Prof. Orientador (UTFPR)

Andreia Scariot Beulke
Banca (UTFPR)

Vinicius Pegorini
Banca (UTFPR)

Robison Cris Brito
Coordenador da IV Especialização em Tecnologia Java

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

DDEBASTIANI, Julio Artur. Servidor REST para emissão de nota fiscal eletrônica. 2017. 27f. Monografia (Trabalho de especialização) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Muitas operações de negócio têm sido realizadas com o auxílio de sistemas computacionais, como os sistemas Enterprise Resource Planning, os de apoio à tomada de decisão e os que realizam simulações, seja com viés científico ou para projeções de investimentos, por exemplo. Esses sistemas podem prover uma automatização ainda maior para os negócios além do controle e do fornecimento de dados para apoio à tomada de decisão. O comércio eletrônico é uma prova efetiva disso. Quando são comercializados itens que podem ser digitais como música e livros, as operações realizadas podem ser puramente digitais: o site oferece os produtos, permite realizar a compra, efetuar o pagamento e provê a entrega do produto (cópia digital do livro ou da música). Outras operações que envolvem a transação física de mercadorias podem beneficiar-se dos sistemas computacionais, além do controle gerencial. Esse adicional é a emissão de notas fiscais eletrônicas para acompanhamento de mercadorias, inclusive no seu transporte. Com esse procedimento a empresa emite um arquivo eletrônico contendo as informações fiscais da operação comercial. Esse arquivo, a Nota Fiscal eletrônica (NF-e), é assinado por meio de um certificado digital, assegurando a integridade dos dados e a autoria do emissor. A NF-e é, então, transmitida para a Secretaria da Fazenda de jurisdição do contribuinte e a partir dela é realizada a autorização eletrônica para o trânsito da respectiva mercadoria. Visando gerar o arquivo da NF-e, por meio deste trabalho foi implementado um servidor REST para emissão desse tipo de nota. A linguagem Java foi a tecnologia principal utilizada para a implementação do aplicativo.

Palavras-chave: NF-e. Servidor REST. Java.

ABSTRACT

DDEBASTIANI, Julio Artur. REST server to emission of electronic invoice. 2017. 27f. Monografia (Trabalho de especialização) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Several business operations have been made with the help of software, such as systems of Enterprise Resource Planning, systems of support for decision-making, and systems to make simulations, scientific simulations (for example, tests of drugs) or business simulations (such as, projection of investments). These software can prove an automation more profitable to the business. Ecommerce is a proof of this. When items that can be digital such as music and books are marketed, the operations performed can be purely digital: the offer of the products, the purchase, the payment and the delivery of the product. Other operations that involve the physical transaction of goods can benefited from software systems, besides the managerial control. This additional is the issuance of electronic invoices for tracking merchandise, including shipping. With this procedure the company emits an electronic file containing the tax information of the commercial operation. This file, the Nota Fiscal Eletrônica (NF-e), is signed by means of a digital certificate, ensuring the integrity of the data and the authorship of the owner. The NF-e is then transmitted to the Secretaria da Fazenda of the taxpayer's jurisdiction and from it the electronic authorization for the transit of the respective merchandise is carried out. In order to generate the file of the NF-e, through this work a REST server was implemented to generate this type of note. The Java language was the main technology used to implement the application.

Keywords: NF-e. REST server. Java.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	19
Figura 2 – Classe movimentacao.....	19
Figura 3 – Classe NFeFacade.....	20

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação do aplicativo	13
Quadro 2 – Requisitos funcionais do sistema.....	15
Quadro 3 – Expansão do caso de uso envio da nota para a Receita Federal.....	16
Quadro 4 – Expansão do caso de uso cancelamento de nota fiscal eletrônica	17
Quadro 5 – Expansão do caso de uso consulta de <i>status</i> do <i>web service</i>	17
Quadro 6 – Expansão do caso de uso criar XML.....	17
Quadro 7 – Expansão do caso de uso consulta assinar XML.....	18
Quadro 8 – Expansão do caso de uso comunicação com o <i>webservice</i>	18
Quadro 9 – Expansão do caso de uso consulta calcular dados do DIFAL.....	18

LISTAGENS DE CÓDIGOS

Listagem 1 – Facade para consulta de status do webservice.....	21
Listagem 2 – Montagem do XM: node infNfe	22
Listagem 3 – Método build() da classe NfeXMLBuilder	23
Listagem 4 – Trecho do XML gerado.....	24

LISTA DE SIGLAS

AC-RFB	Autoridade Certificadora da Receita Federal do Brasil
AR-RFB	Autoridade de Registro da Receita Federal do Brasil
COTEC	Coordenação Geral de Tecnologia e Segurança da Informação
DANFE	Documento Auxiliar da Nota Fiscal Eletrônica
DANFE	Documento Auxiliar de Nota Fiscal Eletrônica
DIFAL	Diferencial de Alíquota
e-CNPJ	Certificado digital de Cadastro Nacional de Pessoas Jurídicas
e-CPF	Certificado digital de Cadastro de Pessoas Físicas
ENCAT	Encontro Nacional de Coordenadores e Administradores Tributários Estaduais
HTTP	<i>Hypertext Transfer Protocol</i>
ICMS	Imposto sobre Circulação de Mercadorias e Serviços
ICP-Brasil	Infraestrutura de Chaves Públicas Brasileira
ID	Identificador
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Tokens</i>
NFCe	Nota Fiscal do Consumidor Eletrônica
NF-e	Nota Fiscal Eletrônica
NFSE	Nota Fiscal de Serviço Eletrônica
REST	<i>Representational State Transfer</i>
RF	Requisitos Funcionais
RFB	Receita Federal do Brasil
SAT	Sistema Autenticador e Transmissor de Cupons
SGBDOR	Sistema Gerenciador de Banco de Dados Objeto Relacional
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	10
2 FERRAMENTAS, TECNOLOGIAS E PROCEDIMENTOS	13
2.1 FERRAMENTAS E TECNOLOGIAS	13
3 RESULTADOS	14
3.1 ESCOPO DO SISTEMA.....	14
3.2 MODELAGEM DO SISTEMA.....	14
3.3 IMPLEMENTAÇÃO DO SISTEMA	21
4 CONSIDERAÇÕES FINAIS	25
REFERÊNCIAS	27

1 INTRODUÇÃO

O desenvolvimento tecnológico associado ao avanço de recursos relacionados à segurança de dados transmitidos e armazenados e à autenticidade desses dados tem feito com que muitas operações de negócio sejam realizadas com auxílio de, ou apenas por meio digital. A compra, o pagamento e a entrega de músicas, filmes e livros, por exemplo, podem ser realizados somente por meio digital.

Embora muitas operações de comércio ainda sejam realizadas com a emissão de papéis para comprovar a sua realização, é natural, pelos aspectos indicados no parágrafo anterior, que seja crescente o número dessas operações sendo realizadas de maneira digital. A emissão de nota fiscal é uma dessas operações que apresenta a possibilidade de emissão e controle por meio digital.

A Nota Fiscal Eletrônica (NF-e) tem como objetivo prover a alteração da sistemática atual de emissão da nota fiscal em papel por nota fiscal eletrônica com validade jurídica para todos os fins (NFE, 2017, p.1). O desenvolvimento da NF-e foi coordenada pelo Encontro Nacional de Coordenadores e Administradores Tributários Estaduais (ENCAT) e desenvolvida em parceria com a Receita Federal do Brasil.

A empresa que emite uma NF-e gera um arquivo eletrônico contendo as informações fiscais da operação comercial, que deve ser assinado por meio de certificado digital para garantir a integridade dos dados e a autoria do emissor. Esse arquivo eletrônico, que corresponde a NF-e é transmitido para a Secretaria da Fazenda de jurisdição do contribuinte. Essa Secretaria faz uma pré-validação do arquivo e devolve um protocolo de recebimento (autorização de uso), sem o qual não poderá haver o trânsito da mercadoria (NFE, 2017, p.1).

A NF-e também é transmitida para a Receita Federal, que é o repositório nacional de todas as NF-e emitidas (ambiente nacional). No caso de operações interestaduais o envio é realizado para a Secretaria de Fazenda de destino da operação e Suframa, quando as mercadorias são destinadas às áreas incentivadas. As Secretarias de Fazenda e a Receita Federal do Brasil disponibilizam consulta pela Internet de dados das NF-e para o destinatário e outros legítimos interessados, que detenham a chave de acesso do documento eletrônico (NFE, 2017, p.1).

Para acompanhar o trânsito da mercadoria é impressa em única via uma representação gráfica simplificada da NF-e, intitulada Documento Auxiliar da Nota Fiscal Eletrônica (DANFE). Esse documento conterà, em destaque, a chave de acesso para consulta da NF-e e

um código de barras bi-dimensional para a confirmação de informações da NF-e pelas unidades fiscais (NFE, 2017, p.1).

O contribuinte destinatário, não emissor de NF-e, poderá escriturar os dados contidos no DANFE para a escrituração da NF-e, sendo que sua validade ficará vinculada à efetiva existência da NF-e nos arquivos das administrações tributárias envolvidas no processo, comprovada pela emissão da autorização de uso. O contribuinte emissor da NF-e realiza a escrituração a partir das NF-e emitidas e recebidas (NFE, 2017, p.1).

O certificado digital, utilizado na assinatura da NF-e, é uma assinatura com validade jurídica que garante proteção às transações eletrônicas e outros serviços via Internet, permitindo que pessoas e empresas se identifiquem e possam realizar assinaturas digitais com mais segurança e agilidade (SERASA, 2017).

A emissão, renovação e revogação de Certificado Digital e-CPF ou e-CNPJ é realizada por uma empresa devidamente autorizada pela Receita Federal do Brasil, denominada Autoridade Certificadora Habilitada. Existem várias autoridades certificadoras habilitadas pela Receita Federal do Brasil, além da própria Receita Federal.

Para uso de Certificado Digital e-CPF ou e-CNPJ, para o correto funcionamento dos certificados e serviços é necessário o uso do navegador Microsoft Internet Explorer, versão 5.50 ou posterior. E o navegador deve estar habilitado para gravação de *cookies* (SUBSECRETARIA..., 2016).

Alguns conceitos básicos relacionados ao Certificado Digital (SUBSECRETARIA..., 2016):

a) Assinatura digital – é o processo eletrônico de assinatura, baseado em sistema criptográfico assimétrico, que permite ao usuário usar sua chave privada para declarar a autoria de documento eletrônico, assegurando a integridade do conteúdo.

b) Autoridade Certificadora da Receita Federal do Brasil (AC-RFB) - é a entidade integrante da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil), responsável pela assinatura dos certificados das Autoridades Certificadoras Habilitadas.

c) Autoridade Certificadora Habilitada - é a entidade integrante da ICP-Brasil em nível imediatamente subsequente ao da AC-RFB, habilitada pela Coordenação Geral de Tecnologia e Segurança da Informação (COTEC), em nome da Receita Federal do Brasil (RFB), responsável pela emissão e administração dos Certificados Digitais e-CPF e e-CNPJ.

d) Autoridade de Registro da Receita Federal do Brasil (AR-RFB) - é a entidade operacionalmente vinculada à AC-RFB, responsável pela confirmação da identidade dos

solicitantes de credenciamento e habilitação como Autoridades Certificadoras integrantes da ICP-Brasil, em nível imediatamente subsequente ao da AC-RFB.

e) Autoridades de Registro - são as entidades operacionalmente vinculadas à determinada Autoridade Certificadora Habilitada, responsáveis pela confirmação da identidade dos solicitantes dos certificados e-CPF e e-CNPJ para o certificado digital de Cadastro de Pessoas Físicas e Cadastro Nacional de Pessoas Jurídicas, respectivamente.

f) Certificado Digital e-CPF ou e-CNPJ - é o documento eletrônico de identidade emitido por Autoridade Certificadora credenciada pela Autoridade Certificadora Raiz da ICP-Brasil – AC Raiz e habilitada pela AC-RFB, que certifica a autenticidade dos emissores e destinatários dos documentos e dados que trafegam em uma rede de comunicação, assegurando a privacidade e a inviolabilidade desses.

g) Documento Eletrônico – é o documento cujas informações são armazenadas, exclusivamente, em meio eletrônico.

h) ICP-Brasil - é um conjunto de técnicas, práticas e procedimentos a ser implementado pelas organizações governamentais e privadas brasileiras com o objetivo de garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras.

i) Usuário - pessoa física ou jurídica titular de Certificado Digital e-CPF ou e-CNPJ, respectivamente, ou de qualquer outro certificado digital emitido por Autoridade Certificadora não habilitada pela RFB e credenciada pela ICP Brasil.

Considerando esse contexto de NF-e e certificação digital, o sistema desenvolvido como resultado deste trabalho receberá dados no formato *JavaScript Object Notation* (JSON) que serão transformados em um arquivo *eXtensible Markup Language* (XML) que será usado para emissão de Nota Fiscal Eletrônica. O arquivo gerado será assinado digitalmente por meio de certificado digital e estará no formato compatível com a NF-e. O objetivo deste trabalho é, assim, implementar um servidor *Representational State Transfer* (REST) para emissão de nota fiscal eletrônica.

2 FERRAMENTAS, TECNOLOGIAS E PROCEDIMENTOS

Para o desenvolvimento do Servidor REST foi utilizada a linguagem Java. A IDE é o NetBeans na versão 8.1. O Spring foi usado para a criação do *microservice* (microserviço), para o mapeamento das *Uniform Resource Locator* (URL) e o gerenciamento das requisições *Hypertext Transfer Protocol* (HTTP). A persistência dos dados foi realizada utilizando o Hibernate e para o Sistema Gerenciador de Banco de Dados Objeto Relacional (SGBDOR) foi utilizado o PostgreSQL.

2.1 FERRAMENTAS E TECNOLOGIAS

O Quadro 1 apresenta as ferramentas e as tecnologias utilizadas na modelagem e no desenvolvimento do aplicativo.

Ferramenta / Tecnologia	Versão	Disponível em	Aplicação
Java	8	https://www.oracle.com/br/java/index.html	Linguagem para desenvolvimento do servidor <i>Representational State Transfer</i> (REST).
Spring		https://spring.io/	<i>Framework</i> usado no mapeamento das requisições do usuário ao servidor REST.
Hibernate		http://hibernate.org/	<i>Framework</i> usado para mapeamento objeto-relacional para persistência dos dados.
NetBeans	8.1	https://netbeans.org/	<i>Integrated Development Environment</i> (IDE) para desenvolvimento do <i>backend</i> .
PostgreSQL		https://www.postgresql.org/	SGBDOR e banco de dados utilizado.

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação do aplicativo

A implementação resultante da realização deste trabalho é um microserviço. Microserviço é uma abordagem para desenvolver uma aplicação como uma suíte de serviços. Nessa abordagem cada um dos serviços é executando em seu próprio processo e eles se comunicam por meio de mecanismos que, geralmente, é uma API HTTP. Cada um dos serviços implementa pequenas responsabilidades e eles são publicados em produção de maneira independente por processos de *deploy* automatizados. Existe um gerenciamento centralizado mínimo desses serviços, que podem ser escritos em linguagens diferentes e usar tecnologias diferentes para armazenamento de dados (LEWIS, FOWLER, 2014).

3 RESULTADOS

Este capítulo apresenta o resultado da realização do trabalho que é um servidor REST para emissão de Notas Fiscais Eletrônicas.

3.1 ESCOPO DO SISTEMA

O aplicativo desenvolvido é *web* e funcionará em um servidor REST, denominado RESTful Web Service e receberá dados no formato JSON. Esses dados serão transformados em XML que será usado para emissão de Nota Fiscal Eletrônica. O arquivo gerado será assinado digitalmente por meio de certificado digital e estará no formato compatível com a NF-e. Em uma primeira versão do aplicativo não haverá baixa de estoque ou movimentação financeira, somente a geração de XML, a assinatura digital de arquivos e a comunicação com *web service* para envio e recebimento de arquivos XML.

A integração com os outros microsserviços será realizada pelo banco de dados uma vez que ele é compartilhado. Um microsserviço proverá os cadastros básicos como pessoa, item para venda, transportadora, natureza da operação entre outros. Outro microsserviço gerenciará as compras e vendas realizadas. O usuário manipulará essas informações e lançará as compras e vendas no banco de dados. Depois de as informações serem salvas, o usuário terá a opção de emitir a NF-e utilizando uma das compras/vendas já persistidas no banco.

Essa rotina será acessada ao término do cadastro da compra/venda ou por meio de uma janela específica para emissão fiscal. Nessa tela aparecerão todos os documentos, tanto os que foram emitidos quanto os que ainda estão pendentes. O aplicativo não possui interface gráfica. Os testes da aplicação foram realizados utilizando classes de teste na linguagem Java e com programas que fazem comunicação via REST.

3.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a listagem dos Requisitos Funcionais (RF) identificados no sistema.

Identificação	Nome	Descrição
RF 01	Criar XML	Criar arquivo XML compatível para emissão de Nota Fiscal Eletrônica. O arquivo deve ser montado a partir de informações persistidas em um banco de dados e também por JSON.
RF 02	Assinar XML	Assinar arquivo gerado utilizando Certificado Digital tipo A1.
RF 03	Comunicar com <i>web service</i>	Conectar e consumir recursos do <i>web service</i> fornecido pela Receita Federal utilizando o Certificado Digital tipo A1
RF 04	Verificar <i>status</i> do serviço	Verificar o <i>status</i> de serviço do <i>web service</i> e processar o retorno recebido.
RF 05	Emitir Nota Fiscal Eletrônica	Enviar arquivo XML assinado à Receita Federal e processar o retorno recebido.
RF 06	Cancelar Nota Fiscal Eletrônica emitida	Enviar um evento de cancelamento ao <i>web service</i> da Receita Federal e processar o retorno recebido
RF 07	Converter dados salvos no banco em um XML	Converter as informações de compra e venda salvas no banco de dados em um arquivo XML.
RF 08	Calcular dados para <i>tag</i> Diferencial de Alíquota (DIFAL)	Calcular informações para o DIFAL de acordo com as informações de Imposto sobre Circulação de Mercadorias e Serviços (ICMS) definidas nos produtos.

Quadro 2 – Requisitos funcionais do sistema

A Figura 1 apresenta os casos de uso do sistema. Há apenas um ator chamado Faturamento. Ele é o responsável pelas operações que envolvem a comunicação do sistema com o *web service* da Receita Federal. As tarefas que ele poderá executar são enviar uma venda para transformá-la em uma nota fiscal eletrônica, cancelar uma nota fiscal eletrônica já emitida e verificar o *status* do *web service*.

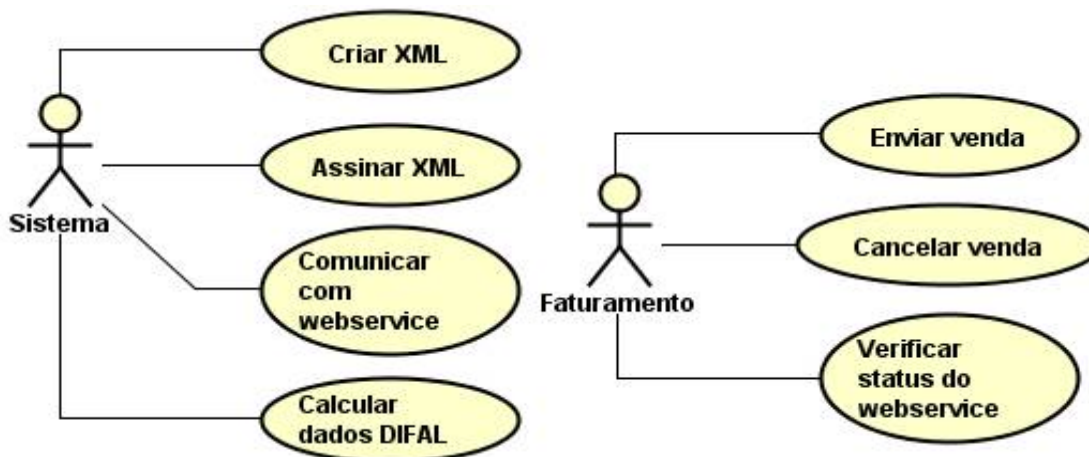


Figura 1 – Diagrama de casos de uso

No Quadro 3 encontra-se a expansão dos casos de uso do faturamento. Esse caso de uso refere-se ao envio da nota para a Receita Federal que é um envio de venda.

<p>1. Identificador do requisito: enviar venda.</p> <p>Descrição: Transformar uma venda que já está persistida no banco de dados em uma nota fiscal eletrônica.</p> <p>Evento iniciador: o ator acessa a URL responsável pelo envio da nota fiscal eletrônica informando o (identificador) ID do registro.</p> <p>Pré-condição: A venda precisa estar salva no banco de dados.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1 – O registro é selecionado a partir do ID informado. 2 – É feita a composição do XML a partir da venda selecionada. 3 – Esse XML é assinado e enviado ao <i>web service</i> correspondente. 4 – O retorno do <i>web service</i> é lido e as informações são persistidas no banco de dados. 5 – A função retorna o <i>status</i> HTTP 200. <p>Pós-Condição: o registro da venda é atualizado com as informações para identificação da nota fiscal eletrônica (chave de acesso, XML assinado) e o <i>status</i> da venda fica como autorizada.</p> <p>Extensões: Se por algum motivo não foi possível enviar a nota fiscal eletrônica o sistema atualizará a venda para o <i>status</i> pendente. O retorno da função será o <i>status</i> HTTP 412.</p>

Quadro 3 – Expansão do caso de uso enviar venda

No Quadro 4 está descrita a funcionalidade de cancelamento da nota fiscal eletrônica.

<p>Identificador do requisito: cancelar venda</p> <p>Descrição: Efetuar o cancelamento de uma nota fiscal eletrônica já emitida.</p> <p>Evento Iniciador: Acesso à URL responsável pelo cancelamento da venda, passando o ID da venda.</p> <p>Pré-condição: a nota fiscal eletrônica deve estar com o <i>status</i> autorizada.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1 – O registro é selecionado no banco de dados. 2 – As informações da nota fiscal são lidas e enviadas ao <i>web service</i> de cancelamento. 3 – O retorno do <i>web service</i> é processado e a venda é atualizada com as informações do cancelamento.

4 – A função retorna o *status* HTTP 200.

Pós-condição: A venda selecionada é atualizada com as informações do cancelamento e a venda fica com o *status* **cancelada**.

Extensões: Se a conexão com o *web service* não foi completada ou o prazo para cancelamento expirou, o *status* HTTP retornado será o 412. O motivo da falha também será informado no retorno.

Quadro 4 – Expansão do caso de uso cancelar venda

O Quadro 5 descreve a funcionalidade de consulta do *status* do *web service*.

Identificador do requisito: verificar *status* do *web service*.

Descrição: O ator poderá consultar o *status* do *web service* do seu estado.

Evento Iniciador: O ator acessa a URL responsável pela consulta de *status*.

Pré-condição: Não há.

Sequência de eventos:

1 – As informações do certificado digital são lidas.

2 – A requisição para o *web service* é feita

3 – O retorno é processado e o *status* é enviado ao usuário.

4 – A função retorna o *status* HTTP 200.

Pós-condição: Dados da consulta apresentados.

Extensões: Se por algum motivo a conexão com o *web bservice* não pode ser realizada, informa o *status* HTTP 412. O motivo também é informado ao usuário.

Quadro 5 – Expansão do caso de uso verificar *status* do *web service*

O Quadro 6 descreve a funcionalidade de criação do XML.

Identificador do requisito: criar XML.

Descrição: Os dados da classe *Movimentacao* serão convertidos em um arquivo XML.

Evento Iniciador: O envio da nota para o *webservice*.

Pré-condição: A classe *Movimentacao* precisa estar persistida no banco de dados.

Sequência de eventos:

1 – A classe *MovimentacaoToNfe* é criada solicitando como parâmetro de criação uma instância da classe *Movimentacao*.

2 – Os atributos classe *Movimentacao* são percorridos e convertidos na classe *Nfe*, que será a responsável por criar o XML.

Pós-condição: A classe *Nfe* estará preenchida com os dados da classe *Movimentacao*.

Extensões: Se alguma informação obrigatória estiver faltando na classe *Movimentacao*, uma exceção será lançada no processo, devendo ser tratada pela classe que iniciou o processo.

Quadro 6 – Expansão do caso de uso criar XML

O Quadro 7 descreve a funcionalidade da assinatura do XML.

Identificador do requisito: assinar XML

Descrição: A classe *Nfe* exportará uma instância da classe *NfeXMLBuilder* que por sua vez exportará o XML que deverá ser assinado.

Evento Iniciador: A classe *NfeFacadeEnvio* será criada e iniciará o processo.

Pré-condição: A classe *Nfe* deve existir e apresentar dados válidos.

Sequência de eventos:

1 – A classe *NfeFacadeEnvio* criará uma instância da classe *NfeXMLBuilder* passando como parâmetro uma instância da classe *Nfe*.

2 – A classe *NfeXMLBuilder* transforma a *Nfe* em uma classe *Document* que representa o XML.

3 – Essa classe é assinada usando um certificado digital válido.
Pós-condição: O arquivo documento deve conter os nodes que representam a assinatura.
Extensões: Caso o certificado digital esteja vencido ou inválido por algum motivo a classe jogará uma exceção que deverá ser tratada por quem a chamou.

Quadro 7 – Expansão do caso de uso consulta assinar XML

O Quadro 8 descreve a comunicação com o webservice da receita federal.

Identificador do requisito: comunicar com webservice.
Descrição: A requisição é feita através de um XML que possui cabeçalho e corpo. A informação contida nessas áreas varia de acordo com a operação que é feita. Para cada operação existem classes especializadas em criar o cabeçalho e o corpo.
Evento Iniciador: A classe NfeFacade é criada e ela chamará a comunicação com o webservice.
Pré-condição: É necessário possuir um certificado digital válido e as informações mínimas para consumir o *webservice*. Essas informações mínimas depende da operação que será feita.
Sequência de eventos:
 1 – O cabeçalho da requisição é criado passando sempre a UF do usuário e a versão da Nfe.
 2 – O corpo da requisição é montado, dependendo da operação que será feita.
 3 – A requisição é feita, passando as duas partes.
 4 – O retorno é lido e apresentado.
Pós-condição: As informações recebidas do *webservice* são lançadas nas classes correspondentes.
Extensões: Se o certificado digital não for válido ou se o *webservice* estiver indisponível no momento do consumo uma exceção será jogada e tratada pelo criador da classe.

Quadro 8 – Expansão do caso de uso comunicar com webservice

O Quadro 9 descreve o cálculo do DIFAL.

Identificador do requisito: calcular dados do DIFAL.
Descrição: A classe Difal monta a tag *icmsUfDest* se for necessário. Verificando os dados da classe *Movimentacao* verifica-se a necessidade de criação dessa tag.
Evento Iniciador: No momento que a classe *Movimentacao* está sendo transformada na classe *Nfe*.
Pré-condição: A UF do emitente deve ser diferente da UF do destinatário, a classificação do destinatário deve ser Não Contribuinte e a CFOP utilizada na nota não pode ser de remessa ou retorno.
Sequência de eventos:
 1 – Os detalhes (produtos) da classe *Movimentacao* são percorridos.
 2 – As informações de ICMS dos produtos são lidas e transformadas nas tags filhas da *icmsUfDest*.
 3 – A soma dessas informações são lançadas na tags de totalização da *Nfe*.
Pós-condição: Se a classe retornou valores válidos a tag *icmsUfDest* é preenchida. Caso contrário essa tag é enviada em branco.

Quadro 9 – Expansão do caso de uso consulta calcular dados do DIFAL

A classe que será transformada em uma Nota Fiscal Eletrônica é a classe *Movimentacao*. Ela será persistida no banco de dados e transformada em *Nfe* no momento que a emissão for iniciada. A Figura 2 apresenta o diagrama dessa classe.



Figura 2 – Classe movimentacao

Outras classes estão ligadas a ela e farão parte do XML. A classe Pessoa, que armazena as informações do destinatário da nota. A classe MovimentacaoImpostos com as informações de ICMS, IPI, ISS, entre outros impostos. A classe MovimentacaoInfoNFe responsável pelas informações da nota fiscal eletrônica quando essa for emitida, como chave de acesso, data de emissão, entre outros. A classe MovimentacaoDetalhe com as informações dos produtos que serão usados na Nfe. A classe que faz o gerenciamento das conexões ao *web services* é a classe NFeFacade. Essa classe é apresentada na Figura 3.

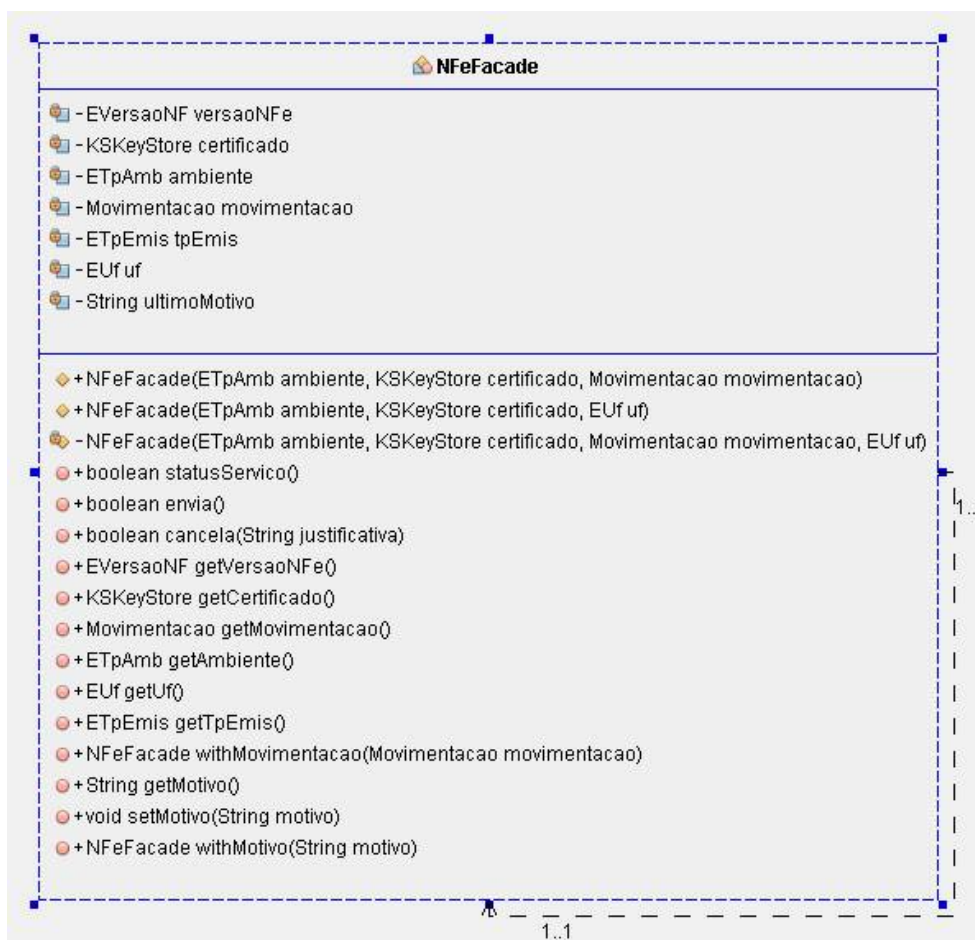


Figura 3 – Classe NFeFacade

Devido à complexidade das operações, alguns métodos chamam outras classes que gerenciam as conexões e o retorno. Um exemplo é o método *envia()* que cria a classe NFeFacadeEnvio. A classe KSKeyStore é a responsável pelo gerenciamento do certificado digital.

3.3 IMPLEMENTAÇÃO DO SISTEMA

Nessa seção são apresentados exemplos dos códigos-fonte que foram implementados no desenvolvimento do aplicativo. Devido à complexidade na troca de arquivos XML entre o aplicativo e o *web service* da Receita Federal todas as operações para comunicação com o *web service* foram incluídas dentro de uma classe que implementa o padrão de projeto *facade*, visando simplificar a criação e a leitura dos arquivos XML.

O código apresentado na Listagem 1 é utilizado para consultar o *status* do *web service*.

```

public boolean consulta() throws WebServiceNotImplementedException, IOException,
    JAXBException, SOAPException {
    NfeStatusServico3Client client = new NfeStatusServico3Client(
        WebServiceInfoFactory.newInstance(
            fcd, EServicoNF.NFE_STATUS_SERVICO
        ), this.fcd.getCertificado()
    );
    RetConsStatServ ret = ( RetConsStatServ ) client.doRequest(
        this.getCabec(), this.getDados()
    );
    System.out.println( ret );
    this.xMotivo = ret.getxMotivo();

    return ret.getcStat().equals( "107" );
}

public String getxMotivo() {
    return xMotivo;
}

private NfeCabecMsgStatusServico getCabec() {
    NfeCabecMsgStatusServico result = new NfeCabecMsgStatusServico();
    result.setCUF( String.valueOf( this.fcd.getUF().getCodigo() ) );
    result.setVersaoDados( this.fcd.getVersaoNFe().getVersao() );
    return result;
}

private NfeDadosMsgStatusServico getDados() throws JAXBException {
    NfeDadosMsgStatusServico result = new NfeDadosMsgStatusServico();
    result.getContent().add(
        GenericMarshaler.getDocumentElement( ConsStatServ.class,
            new ConsStatServ(
                this.fcd.getVersaoNFe().getVersao(),
                String.valueOf( this.fcd.getAmbiente().getId() ),
                String.valueOf( this.fcd.getUF().getCodigo() )
            )
        )
    );
    return result;
}

```

Listagem 1 – Facade para consulta de status do webservice

Várias classes são usadas no trecho de código da Listagem 1. A classe *NfecabecMsgStatusServico* é usada para montar o cabeçalho da requisição. Nela são informados o código da Unidade Federativa e a versão da NF-e. Na classe *NfeDadosMsgStatusServico* o corpo da mensagem é montado. Nele são enviados a versão da NF-e, o ambiente (homologação ou produção) e o código da Unidade Federativa. Essas duas classes são usadas na requisição que é realizada pela *NfeStatusServico3Client*. O retorno dessa classe é um objeto da classe *RetConsStatServ*. Nela está o Código de Status (cStat) e o Motivo (xMotivo).

A montagem do XML é realizada pela classe *Nfe*. Ela possui estrutura igual ao XML esperado pelo *web service*, sendo que os atributos são os *nodes*. Ao receber o JSON com a venda feita, seus dados são usados para preenchimento da *Nfe*. Quando todos os atributos foram preenchidos corretamente, a classe é transformada em um arquivo XML.

A Listagem 2 apresenta partes de como é o processo da montagem do XML que é enviado ao *web service*.

```
@XmlElement( name = "ide", required = true )
private Ide ide;
XmlElement( name = "emit", required = true )
private Emit emit;
XmlElement( name = "avulsa", required = false )
private Avulsa avulsa;
XmlElement( name = "dest", required = false )
private Dest dest;
XmlElement( name = "retirada", required = false )
private Retirada retirada;
XmlElement( name = "entrega", required = false )
private Entrega entrega;
XmlElement( name = "autXML", required = false )
private List<AutXML> autXMLList;
XmlElement( name = "det", required = true )
private List<Det> detList;
```

Listagem 2 – Montagem do XM: node infNfe

Para criar o arquivo apresentado na Listagem 2 foi utilizada a anotação *@XmlElement*. Ela serve para mapear os atributos da classe que serão convertidos em texto posteriormente. A conversão é feita *NfeXMLBuilder* no método *build()* de acordo com o código que está na Listagem 3.

```

private void build() throws JAXBException, XMLStreamException,
    ParserConfigurationException, SAXException, IOException {
    this.doc = GenericMarshal ler.getDocumentElementWithoutNS(
        Nfe.class, nfe ).getOwnerDocument();
}

```

Listagem 3 – Método build() da classe NfeXMLBuilder

Passando a classe Nfe e o objeto como parâmetros, a função percorre todos os atributos da classe que estão anotados com @XmlElement transformando-os em atributos para o objeto *doc* que é do tipo *Document*. Esse *Document* depois é assinado pela classe *NfeXMLAICertificateSignatureKeyStore*.

Com relação à segurança do microserviço, ela é feita utilizando *JSON Web Tokens* (JWT). Qualquer cliente que deseje fazer alguma requisição deve informar um *token*. Esse *token* é validado e, caso seja inválido, retornará o status HTTP 401.

Esse *token* é recebido no momento que o usuário logar no sistema, passando as informações de controle (CNPJ, usuário, senha entre outras). A grande maioria das requisições passa por essa validação.

O projeto é um microserviço que funciona em conjunto com outros. Para fazer o gerenciamento das dependências, foi utilizado o *Maven*. Todas as dependências comuns entre os projetos ficam no arquivo *pom.xml* do projeto principal. As dependências específicas de cada projeto ficam no arquivo *pom.xml* específico.

A Listagem 4 apresenta um trecho do XML gerado.

```

<i de>
  <CUF>35</CUF>
  <CNF>00000715</CNF>
  <natOp>VENDA</natOp>
  <i ndPag>0</i ndPag>
  <mod>55</mod>
  <serie>1</serie>
  <nNF>715</nNF>
  <dhEmi >2017-03-02T09:00:00-03:00</dhEmi >
  <dhSai Ent>2017-03-02T09:00:00-03:00</dhSai Ent>
  <tpNF>1</tpNF>
  <i dDest>1</i dDest>
  <cMunFG>3550308</cMunFG>
  <tpImp>1</tpImp>
  <tpEmis>1</tpEmis>
  <cDV>2</cDV>
  <tpAmb>2</tpAmb>
  <fi nNFe>1</fi nNFe>

```



```

<i ndFi nal >1</i ndFi nal >
<i ndPres>1</i ndPres>
<procEmi >3</procEmi >
<verProc>0. 1</verProc>
</i de>
<emi t>
  <CNPJ>00999999000999</CNPJ>
  <xNome>Teste Prod e Serv ME</xNome>
  <xFant>Teste Prod e Serv ME</xFant>
  <enderEmi t>
    <xLgr>Rua Teste</xLgr>
    <nro>10</nro>
    <xBai rro>Teste</xBai rro>
    <cMun>3550308</cMun>
    <xMun>Sao Paul o</xMun>
    <UF>SP</UF>
    <CEP>04444000</CEP>
    <cPai s>1058</cPai s>
    <xPai s>BRASI L</xPai s>
    <fone>1112345678</fone>
  </enderEmi t><I E>113111113111</I E>
  <I M>36993699</I M>
  <CNAE>4751021</CNAE>
  <CRT>1</CRT>
</emi t>
<dest>
  <CPF>00009009000</CPF>
  <xNome>Cl i ente Teste</xNome>
  <enderDest>
    <xLgr>Rua Outro Teste</xLgr>
    <nro>99</nro>
    <xCpl >apartamento 302</xCpl >
    <xBai rro>Bai rro teste</xBai rro>
    <cMun>3550308</cMun>
    <xMun>Sao Paul o</xMun>
    <UF>SP</UF>
    <CEP>01111000</CEP>
    <cPai s>1058</cPai s>
    <xPai s>BRASI L</xPai s>
    <fone>115554444</fone>
  </enderDest>
  <i ndI EDest>9</i ndI EDest>
  <emai l >emai l @emai l . com. br</emai l >
</dest>

```

Listagem 4 – Trecho do XML gerado

4 CONSIDERAÇÕES FINAIS

O objetivo desse trabalho foi construir um microsserviço usando REST que pudesse fazer a conversão de vendas cadastradas em um banco de dados em arquivos XML que seguem o padrão da nota fiscal eletrônica, cancelar essas notas e consultar o *status* do *web service* da Receita Federal.

Uma das dificuldades encontradas foi a criação do arquivo XML. A documentação fornecida pela Receita Federal é bem completa, o que facilitou a implementação das classes que fariam esse trabalho. Contudo, o XML que deve ser montado é bem extenso, contendo muitos *nodes*. Isso deixou as classes com muitas linhas de código e como é natural para a linguagem Java a escrita de bastante código, isso se tornou um ponto preocupante no projeto. A anotação `@XmlElement` facilitou muito o trabalho, porque ao usá-la é dispensado o uso dos métodos *getters* para leitura dos atributos

Com relação à comunicação com o *web service* o assistente de importação criou corretamente as classes para criação do cabeçalho e do corpo das requisições. No entanto, os nomes das classes ficaram iguais. Em vários momentos isso ocasionou problemas, porque para iniciar a comunicação é necessário montar uma requisição com cabeçalho e corpo. Como as classes possuíam os mesmos nomes, foi difícil saber quais deveriam ser utilizadas em conjunto para gerar essas duas informações, mesmo estando em pacotes distintos. A solução encontrada foi renomear as classes para nomes que mais facilmente permitissem identificar o seu objetivo e o relacionamento com o projeto.

Um último contratempo foi o vencimento do certificado digital. Infelizmente a Receita Federal não disponibiliza certificados para desenvolvedores, sendo necessária a utilização de certificados reais. Durante o desenvolvimento do projeto o certificado que estava sendo utilizado expirou, impossibilitando a comunicação com o *web service* da Receita Federal. Toda a parte de assinatura do XML e comunicação já haviam sido implementadas e devidamente testadas, porém não foi possível realizar mais testes nas rotinas que envolvem essas operações.

Esse projeto ainda está no início, sendo que falta implementar várias outras funções do *web service* que são necessárias para a emissão fiscal, como Envio de Carta de Correção, Inutilização de NF-e, Impressão de Documento Auxiliar de Nota Fiscal Eletrônica (DANFE), entre outros. Também está prevista a integração com outros serviços, como Nota Fiscal do

Consumidor Eletrônica (NFCe), Sistema Autenticador e Transmissor de Cupons (SAT), e Nota Fiscal de Serviço Eletrônica (NFSE).

REFERÊNCIAS

LEWIS, James; FOWLER, Martin. **Microservices: a definition of this new architectural term**. 2014. Disponível em: <<https://martinfowler.com/articles/microservices.html>>. Acesso em: 20 out. 2017.

NFE. **Nota Fiscal Eletrônica**. Disponível em: <<https://www.nfe.fazenda.gov.br/portal/principal.aspx>>. Acesso em: 02 jan. 2017.

SERASA. **O que é certificado digital**. Disponível em: <<https://serasa.certificadodigital.com.br/o-que-e/>>. Acesso em: 02 jan. 2017.

SUBSECRETARIA DE GESTÃO CORPORATIVA. **Orientações sobre Emissão, Renovação e Revogação de Certificados Digitais e-CPF ou e-CNPJ**. 2016. Disponível em: <<http://idg.receita.fazenda.gov.br/orientacao/tributaria/senhas-e-procuracoes/senhas/certificados-digitais/orientacoes-sobre-emissao-renovacao-e-revogacao-de-certificados-digitais-e-cpf-ou-e-cnpj>>. Acesso em: 02 jan. 2017.